

Readme

Design of indexer

The indexer uses a specially designed data type, comprised of three parts: A index object, a word object, and a file object. The file object contains a string for a filename, an integer for storing the count of the word in that file, and a pointer to the next node for a linked list. The word object contains a string for the word, a pointer to a file node for a file linked list, a pointer to a word for a linked list of words, and an int for total count – the individual counts are stored by the list of file nodes. The index object contains a list of words.

Analysis of time and space usage

Time:

Searching and inserting take $O(n)$ time for words and files.

Space:

Structures use $O(n*k)$ space for n files with k terms each (worst case: all files have different words; best case: all files have the same k words $O(1*k)$)